(2)

UW/NW VLSI CONSORTIUM

Semiannual Technical Report No. 5

University of Washington

October 22, 1984

Reporting Period: 27 March 1984 to 22 October 1984
Principal Investigator: Lawrence Snyder

Sponsored by
Defense Advanced Research Projects Agency (DoD)
ARPA Order No. 4563
Issued by Defense Supply Service-Washington
Under Contract # MDA903-82-C-0424

Contents

1 Executive Summary

1.1 Scope of This Report

This document reports on the research activities of the University of Washington / Northwest VLSI Consortium for the period 26 March 1984 - 22 October 1984 under the sponsorship of the Defense Advanced Research Projects Agency.

1.2 Objectives

The goal of the UW/NW VLSI Consortium is to pursue research, development and education in the use of computer aided design tools for VLSI systems with emphasis on technology transfer.

1.3 Accomplishments

The reporting period has been a very active time for the VLSI Consortium. The primary accomplishments ~~are:~~ for the period Mar. 26 - Oct. 22, 1984 were:

1. CMOS Release. The Consortium began in August to ship copies of its CMOS Design System to universities and government laboratories. The Design System is a reasonably complete collection of tools written by the Consortium or gathered from other universities — University of California - Berkeley, Carnegie-Mellon University and Massachusetts Institute of Technology, etc. — that have been made to function compatibly. In addition to promising full support of CMOS technology, this release dramatically improves on our previous nMOS Release by providing extensive tutorial material for new users and teachers.

2. New DRC. In an effort to improve the performance of VLSI design rule checkers (DRC), the Consortium began a comparative analysis between the algorithms used in the Carnegie-Mellon DRC and a new algorithm developed at the University of Washington. The experiments used a variety of chips including the RISC II and PSC (progammable systolic chip). The results showed that the new approach gives improved performance and the Consortium will add this new DRC to future releases, and

3. Chip designs. In this reporting period, several chip designs have been sent to MOSIS for fabrication. We have done CMOS designs for a CHiP Computer Switch, a ROM, a Multiplier, and two runs of our standard cells for the CMOS Release. We have also fabricated an nMOS processor element for a Pyramid architecture. Finally, we have fabricated CMOS class projects through our member firm, Microtel Pacific Research.

4. Consortium Formation Workshops. The University of Washington/Northwest VLSI Consortium is a cooperative arrangement between the University, DARPA, and five

Northwest high technology companies: Boeing Aerospace, Fluke Manufacturing, Honeywell Marine Systems, Microtel Pacific Research, and Tektronix. Since we are a unique entity, we have participated in presentations on the topic of "Mechanisms for Technology Transfer". First, we participated in a workshop at the Twenty-first Annual Design Automation Conference. Second, we held a workshop on Consortium Formation at the University of Washington. Both opportunities allowed us to promote what we believe is a very effective mode of cooperation.

2 The CMOS Release

Perhaps the most important goal driving our first DARPA contract period has been to amass VLSI design software from universities and other sources, and to package it into a coherent and useful form. This work required modification of other's software as well as the creation of our own. Our first effort was a design system to support the nMOS technology, released in October 1983. In August 1984, we released our CMOS design system. This system not only supported the CMOS technology (as well as nMOS) but it was both more complete in its set of tools, and more thorough in that it contained copious tutorial information. To date we have received 102 requests for the nMOS release and 63 requests for the CMOS release.

2.1 Technical Descriptions of Release

The first release(1.0) of the UW/NW VLSI Design System fully supported only nMOS processes. The current design system release(2.0) supports both the GTE 5 micron isoplanar CMOS process and the MOSIS 3 micron bulk CMOS process as well as MOSIS nMOS processes. Technology files are employed wherever possible to isolate technology dependence.

We have made a number of enhancements to the tools since the first release. The pascal layout artwork program PLAP can now write Manhattan designs in the Caesar database format, so that low level cell design can be done with caesar and combined with the procedural approach of PLAP. A library of PLAP routines for placement and interconnect of cells as well as larger modules is included.

A new display program will display layouts on terminals supporting Tektronix 4010 vector graphics protocol. This program provides a convenient interface for obtaining a hardcopy through penplotters. It supports the hierarchy of the design and has a windowing capability to improve resolution.

We have modified the programs that use the ".sim" file circuit description, eliminating the inconvenience of having to convert between the MIT and Berkeley formats. Programs that read the ".sim" file, such as CRYSTAL, SIM2SPICE and PRESIM now read either format.

The event-driven simulator RNL (written by Chris Terman of MIT) has been considerably enhanced. Additional primitives have been added so that the RNL interface would be more consistent with most versions of Lisp. A recoding of the Lisp interpreter allowed node status to be changed far more efficiently, resulting in a 10-fold decrease in simulation execution times for large designs. Finally, a patterns package was added so that complex and repetitive digital waveforms could be generated far more easily. The recently written program MTP (Multiple Time-series Program) displays RNL and SPICE waveforms on a Printronix printer.

2.2 Tutorials and Standard Cells

Layouts for a set of standard cells implementing simple logic functions are available in both GTE isoplanar CMOS and MOSIS bulk CMOS. These cells include a clocked inverter; 2,3 and 4 input nand and nor; exclusive or; and D latches. The cells have been thoroughly characterized and documented in the release manual. PLA templates in both CMOS technologies may be used in conjunction with the Berkeley program TPLA to generate PLA's. Fabrications of both the standard cells and the PLA's have shown the designs to be functionally correct.

Because of the complexity of the command syntax of RNL, we have developed several tutorial aids. The "NETLIST and RNL Tutorial for Beginners" takes a 10 bit shift register from inception in NETLIST, to simulation with RNL, and finally to waveform plotting with MTP. The "NETLIST/PRESIM/RNL - A Tutorial" contains advanced features of RNL.

A summary of the programs included in the current release of the tools in contained in the Appendix I.

2.3 Unix 4.2

Release 2.0 is supported under the Berkeley 4.1 distribution of UNIX. Recent modifications to the programs will allow support of UNIX 4.2 and will be distributed as Release 2.1. This release will include the addition of second layer metal capability to the MOSIS CMOS ruleset (for LYRA) as well as the extractor MEXTRA. All of the MOSIS CMOS standard cells have been redesigned to take up less area. Three new cells have been added: a full adder, a full multiplier, and an exclusive nor. The MOSIS CMOS pads (originally developed at MIT) have been conditioned for the CBPEM2 process. Also to be included in this release is a pad frame generator for the MOSIS CMOS. It has reduced the amount of user effort to produce the pad interface for the final chip (from approximately 2 hours to 10 minutes). This generator has been used successfully on 5 chip submissions.

3 A New DRC

One of our main scientific activities has been to investigate the design rule checking

problem for ways to improve the performance of these critical tools. The chief participants have been L. Snyder and M. Subramanian.

3.1 Independent Verification

In 1980, Bentley, Haken and Hon[1] gathered some data about the characteristics of a selected group of nMOS designs. This data revealed a certain consistency that could be exploited in the construction of a design rule checker, since the checking algorithms could be optimized for this particular class of structures. The "CMU DRC"[2] was the result of this approach.

Two questions arose concerning these statistics: First, are they still applicable to nMOS designs, given that the CIF files now produced by such tools as Caesar can bias the frequency of certain shaped rectangles? Second, are the statistics applicable to CMOS? Using recent nMOS designs, including PSC and RISC II, as well as CMOS designs, we checked the Bentley, Haken, Hon statistics. The results, reported in Subramanian's master's thesis, are, generally speaking, suprisingly consistent with the earlier results.

3.2 Swart's Algorithm

The following algorithm is a simplified version of Swart's solution to the union problem.[3] This can be easily adapted to solve the intersection problem.

The algorithm employs a planar sweep technique with dynamic partitioning of bins. The sweep-line is horizontal, stepping through increasing values of Y and the X-axis is divided into a number of bins at each position of the sweep line. The data structure for the bins is a linked list of records, one each for every X-coordinate where an endpoint occurs.

Each record contains information on

- the X-value of its position
- the count of how many endpoints fall at its X-position, i.e. a "repeat-count".
- the number of "regions" to the right of the X-position where a "region" is defined as the area enclosed by a rectangle, including its boundary - call this the "region-count"
- a flag to indicate if a vertical segment to be output is beginning or ending.

---

[1] Jon Louis Bentley, Dorothea Haken, and Robert W. Hon, "Statistics on VLSI Designs," *Technical Report* CMU-CS-80-111, Carnegie-Mellon University, April 1980.

[2] Dorothea Haken, "Design Rule Checking on the VLSI Vax", *Technical Report* V052, Carnegie-Mellon University, June 1980.

[3] Garret Swart and Richard Ladner, "Effective Algorithms for Reporting Intersections, *Technical Report* 83-07-03, University of Washington, July 1983.

When an interval is encountered during the planar sweep the following things have to be done

- If the interal corresponds to the bottom of a rectangle then
  1. Insert left endpoint x_1 at the appropriate place. Create a new record if one at x_1 does not already exist. If x_1 already exists then increase its repeat-count by one.
  2. Traverse the chain to the right. Output all segments that entirely lie outside any regions. This could be done by checking the "region-count" of the records that one passes. When outputting a horizontal segment the vertical output flag is used to determine if a vertical segment is to be output.
  3. Insert right endpoint x_2 at the appropriate place. As before if the record already exists its repeat-count is incremented by 1.
- If the interval corresponds to the top of a rectangle then
  1. Delete the reocrd corresponding to left endpoint if the repeat-count is one. If the repeat-count is greater than one then reduce it by one and reduce region-count by one.
  2. Traverse the chain link until the right endpoint, reducing region-count along the way. Output horizontal and vertical segments in much the same way as discussed during the discussion on handling the bottom edge.
  3. Delete the record corresponding to the right endpoint if repeat-count is one. If repeat-count is greater than one then reduce it by one and the region-count of such a record remains the same.

Thus an interval would never by checked against another unless there is some overlap between the two. Keeping a region-count is an efficient way of finding parts of an interval that do not lie inside any interval. One starts with the left endpoint and merely has to traverse the linked list to the position of the right endpoint to generate all segments needed in the output.

Generating output in the order that segments belonging to the union of rectangles are found assures a set sorted by *hiy*. As segments can be output as and when they are found no extra storage is needed to store them either. There is also just one extra sorting task to be done before this set of rectangles can be an input to another planar sweep routine. this of course is to sort them by *loy*.

The main problem in maintaining the linked list of endpoints is of course insertion and deletion. Given an endpoint x, how does one get to the corresponding record as quickly as possible. Clearly traversing the chain from one end to the desired point would make the search very inefficient. Swart's solution to this was to maintain an array of pointers

$A[1 : M]$ such that $A[I]$, would point to the record x_j where x_j is the least X-value >= $C * I$. This kind of a hashing scheme ensures quick access to any record. The efficiency of this scheme is critically dependent on the constant C. Choosing C to be the average edge length in much the same way as the static partitioning algorithm helps in the follwoing way. If C is the average edge length then on the average one has to traverse one or two records before reaching the desired point along the chain. Thus we get constant access time on the average.

It is clear that this approach has a very simple but effective data structure that makes efficient checking of intersections possible. The author implemented this algorithm and grafted it on the DRC. The experimental data gathered in that process and more details on the relative performance of these algorithms are presented in the thesis.

4 Chip Designs

The chip design work that we have engaged in has come in two forms: Chips designed for internal Consortium projects, and chips designed as a result of class work. (The latter designs are fabricated through MPR and are discussed in Section 6.) We give a brief account of the more significant chips.

4.1 Switch Chip

The CHiP Computer has special circuit switches as one of its key components. Because of this importance, test versions of these switches have been built with nMOS. During the reporting peoriod, Wayne Winder built a CMOS version of a 16 switch tool array. The design of the chip not only gave us a chance to compare the performance of the two technologies, it gave us a chance to assess the "conversion" activity of moving between the two technologies. As experienced CMOS designers will understand, the degree to which an nMOS chip can guide the CMOS design is considerably less than one would hope. The performance comparisons between the two chips are still pending changes to our test facilities.

4.2 Preliminary Designs for Generators

We have begun our VLSI Design Generator Project by developing circuit designs that will one day be converted into production generators. Our goal at this point is to experiment with design alternatives and to gather data to be incorporated into the generator "model" of each circuit.

4.2.1 The ROM

A ROM design was undertaken that could be easily encapsulated in a generator, al-

lowing ROMs of an arbitrary aspect ratio (heigth/width), size and relative speed to be created automatically. The following assumptions are imbedded in the design.

- CMOS technology:
  - Devices which interface between functional blocks should be static (isolate charge sharing problems, good rising and falling drive).
  - CMOS static gates only consume power during transitions.
- Mostly NMOS design:
  - n-channel transistors have lower on resistance per unit area.
  - Isolate all n-channel transistors in a single P-well. (P-well to P+ in n-substrate spacing inefficiencies are minimized)
- Precharged NOR form logic:
  - Fast and small as no transistors are in series.
- Transistor active area orientations:
  - In order to allow transistors of arbitrarily low ON resistance to be created (without affecting adjacent cells), the path of current flow through the active area is at right angles to the direction of unrestricted growth. This facillitates trading area for speed.
- Sense Amplifier:
  - The bit transistor is as small as possible to allow large ROMs, unfortunately each bit transistor is asked to discharge a larger and larger capacitance as the ROM grows, slowing operation. This design alleviates this problem by using a smaller voltage change to indicate the presence of a bit transistor (with a sense amplifier).
- Self-Timed:
  - Clocked designs place stringent requirements on the quality of the clocks (rise/fall times,clock skew), and it can be very difficult to meet these needs while distributing clocks over long distances. This design copes with these problems by generating its own clocks with self-timing circuitry. The only restriction is that other modules can not query the ROM more frequently than its cyle time.

This design is very dense owing to the fact that wherever possible, power is provided through signal traces. A rough estimate indicates that a 16Kbyte ROM can be fabricated on the larger MOSIS standard frames. Simulations indicate a cycle time of approximately 300nSec for the larger instances.

We have currently submitted two designs to MOSIS (32x2 bits and 128x8 bits). Neither submittal contains the self-timing circuitry (they require external clocks). This was done to get some information about the basic designs' scaling properties before proceeding with more exotic refinements. Both submittals use inverters with asymetrical pull-up and pull-down transistors to implement the sense amplifiers. These will be replaced with cross-coupled sense amplifiers on subsequent submittals.

4.2.2 The Multiplier

We have investigated various designs for implementing CMOS multipliers. We have submitted two multiplier chips for fabrication, an 8 by 8 unsigned multiplier and an 8 by 8 signed (two's complement) multiplier. These multipliers use the standard "sum diagonal right, carry down" approach. The multiplier operand (n bits) is input in parallel on the left, lsb on the top. The multiplicand operand (m bits) is input in parallel on the top, lsb on the right. The low-order n bits of the product are output on the right, lsb on the top. The high-order m bits of the product are output on the bottom, msb on the left. In this approach, a rectangular array (m by n) of full, one-bit multiplier cells is created. Each cell forms the partial product of the appropriate bit of the multiplicand and the appropriate bit of the multiplier and adds the bit formed with the appropriate sum and carry bits from the previous row. The sums formed are passed down one row and right one column, the carries formed are passed down one row. (The technique is quite similar to manual multiplication with additions occuring after each row.) The bit pairs at the bottom of this array are then summed with a ripple-carry adder (m-1 cells wide) to produce the high-order m bits of the product. The signed multiplier is an extension of this topology to account for the negative values of the most significant bits in the multiplier and the multiplicand (and, of course, the product).

We are using the knowledge gained from the implementation of the chips being fabricated to create a CMOS multiplier generator. This generator will have a leaf-cell assembly approach. The leaf-cells have been designed, work is proceeding on the assembler. The generator will create m by n unsigned or signed (two's complement) multipliers. We intend to pursue multipliers further by implementing 2-bit Booth encoding and conditional sum addition (to replace the ripple-carry adder) to enhance performance (speed). The aim is to have these options available in the multiplier generator.

4.2.3 The Pyramid Processor Element

Professor Steven Tanimoto, of the University of Washington Computer Science Department, designed a chip with Terry Ligocki, a graduate student, called the "Hierarchical

Cellular Logic" (HCL) chip. The chip contains 16 processing elements to be used on 1-bit data in an SIMD fashion. These processing elements are then to be composed into a cellular array computer intended for use in image processing. The HCL chip was fabricated in nMOS on September 6, 1984; it uses a 64 pin package, roughly one third of a standard die, and has about 10K transistors.

5 Consortium Formation Workshops

The Consortium has been involved in two workshops on the topic of technological cooperation. Both have provided an opportunity to present to the technical community the unique aspects of the Consortium's structure.

5.1 Workshop at the DAC

The Consortium was invited to present a workshop at the Twenty First Design Automation Conference held in Albuquerque, New Mexico, on June 25-27, 1984. Entitled *A Model for University/Industry/Government Cooperation*, the workshop attracted an audience of more than one hundred.

The moderator for the session was Paul R. Young, Chairman of the University of Washington Computer Science Department. The participants were: 1) William I. Henry, Manager Electronics Test, Simulation and VLSI CAD, Boeing Aerospace Company, 2) Henriecus Koeman, Director, UW/NW VLSI Consortium, and Liaison for Fluke Manufacturing Company, 3) Robert W. Ritchie, Manager, Computer Science Laboratory, Xerox PARC, 4) Lawrence Snyder, Professor and Principal Investigator, UW/NW VLSI Consortium.

Briefly, the workshop addressed topics such as startup pains, dealing with lawyers, the Consortium's structure, the University's view of the costs and benefits, industry's view of the costs and benefits, the day-to-day operation, and the future directions. It was emphasized that participation in the Consortium furthers everyone's "enlightened self interest" and leverages everyone's contribution greatly.

In a related matter, two of the Workshop's speakers, Young and Henry, participated in a discussion session sponsored by *Design and Test* on the topic of cooperative arrangements for technology transfer. The purpose of the discussion was to gather material for an article in an upcoming issue.

5.2 UW Workshop

In order to analyze the topic of technological cooperation from a somewhat broader perspective, the Consortium sponsored a workshop entitled *"Conference on Technology Based Consortium Formation"*. Organized by Ted Kehl with the help of Vicky Palm and

Merry Bush, the workshop was held at the University Tower Hotel in Seattle, on August 6-7, 1984.

The workshop, composed of an especially distinguished list of speakers, attracted participants from around the country. (Appendix II contains a copy of the program.) Among the highlights were:

- A keynote address by C. Gordon Bell.
- Presentations by Bobby Inman (MCC), Donald Beilman (MCNC), Paul Penfield (MIT), and Larry Sumney (SRC) on their cooperataive structures.
- A discussion of the UW Consortium experience with Ted Kehl, Kit Bradley, Bill Henry, and Paul Losleben.

There was a movie presented by Gwen Bell of the Computer Museum in Marlboro, Massachusetts. Social events included a boat ride on Puget Sound on the Virginia V, and an opportunity to tour the China Exhibit at the Seattle Center.

The total number of attendees was 102. Twenty-six states, with delegates from universities, industries and state government were represented.

From comments on Merry Bush's questionnaire/evaluation form filled out by participants, the conference was very successful. All comments on the meeting's purpose, content and structure were positive and most respondees communicated an interest in the subject matter presented. Of those responding to a question about their attendance at a future meeting, 81% would attend, 15% were unresponsive, and 4% would not attend.

6 The Intensive CMOS Design Class

Again this year an intensive CMOS IC design class was held during the summer. There were 10 participants and the cost, including the fabrication of prototypes using the Isoplanar CMOS process of GTE was $ 4,000 ($ 3,500 for Consortium members).

Lectures were presented during 2 periods of three days in succession; the first period was devoted to MOS theory and electrical design strategy. The three week interval between the first and second periods allowed for the participants to complete an electrical design of their choice and do the electrical simulation. During the last period of three days instruction, lay-out design strategies and tools were introduced. Lab exercises were mixed with the formal lectures to familiarize the participants with the use of the tools for succesful implementation of their own design with a minimum amount of difficulty. All formal lectures were videotaped (a total of 24 hours); tapes can be borrowed from the Consortium library.

The instructors were:

- University faculty for MOS theory and design,

- Consortium staff for the guided lab exercises and
- Industrial lay-out designers for lay-out implementation strategies.

The course outline is given in Appendix III.

The tools used are all in the standard 4.2 release of the UW/NW VLSI Consortium; the main tools used were:

Simulation tools	: RNL for switch level simulation SPICE (very limited use only)
PLA design tools	: PEG, EQNTOTT and PRESTO
Lay-out tools	: CAESAR, PLAP, TPLA and autorouting utilities based on PLAP routines
Design rule verification	: LYRA
Circuit extraction	: MEXTRA

It appeared that class participants had widely varying backgrounds. Some designed very simple circuits and were mainly interested in having at least used every utility in the design system. Others were very aggressive and designed rather sophisticated circuit functions. A significant drawback in this kind of class is that most participants are not relieved from their normal duties for the duration of the class. As a result, the larger projects do not get finished in a timely fashion. Also, since a significant amount of time may elapse between instruction in how to use a tool and actually applying it, several participants tended to use brute force to complete their designs, rather than exploiting the utilities. The simple designs are currently in fabrication and the complex ones close to completion.

Another approach we may follow in the future is to allow a significant amount of self-study using videotapes and books. We have several tutorials developed which will allow engineers to gain sufficient knowledge on the use of tools. We can provide the option of using the system remotely, thereby avoiding travel to the lab. Finally we could make a consultant available to resolve problem areas. We found enough interest to warrant another intensive class, but current responsibilities and other priorities often prevents their participation. A more flexible approach towards educating industrial participants may be more successful.

APPENDIX: I

TOOL DESCRIPTIONS

The following is a brief overview of the vlsi-tools we are distributing. An asterisk (*) appears after the name of tools that are part of the Berkeley Distribution.

The functonal design tools translate high level design descriptions into layout tool input. Layout tools are used to design the actual artwork for the circuit; display tools are used to display circuit designs. Electrical and design rule checkers are included as well as a variety of timing and logic simulation tools.

Functional Design Tools

Translate high level design descriptions into layout tool input.

peg * — Translates a language description of a finite state machine into logic equations compatible with *eqntott*. (Gordon Hamachi, UCB)

eqntott * — Converts logic equations into a truth table format to be used as input to *mkpla* or *tpla*. (Bob Cmelik, UCB)

presto — Tries to minimize the number of product terms in a truth table (UCB).

Layout Tools

Design artwork for circuit.

plap — Compiles a pascal design file created by the user, links it to a package of artwork generation routines, and executes it to produce artwork data files in either db or Caesar formats. Handles general nonmanhattan designs. (Bruce Yanagida, Boeing)

caesar * — A powerful display editor for manhattan designs written at Berkeley. Runs on AED512 or Metheus Omega 440 color displays. Requires a design to be inputted in *caesar* format but will optionally output in CIF. (John Ousterhout, UCB)

tpack * — A collection of C routines that assembles Caesar-generated artwork tiles into semi-regular design modules. This is not an executable program but a library of routines. Employed by *quilt* and *tpla*. (Robert Mayo and John Ousterhout, UCB)

quilt * — Assembles Caesar-generated artwork tiles into a rectangular array. (Robert Mayo, UCB)

tpla * — Technology independent artwork generator. Employs Caesar-generated artwork tiles and a truth table. (Robert Mayo, UCB)

mkpla * — Generates NMOS PLA artwork from a truth table. This program has been largely superceded by *tpla*. (Howard A. Landman, UCB)

Display Tools

Display circuit designs.

cifplot * — Berkeley program that plots a CIF design in stipple patterns on Versatec or Printronix dot-matrix printers. (Dan Fitzpatrick, UCB)

penplot — Penplotting programs for HP 4- and 8-pen plotters. Compatible with either db or CIF formatted designs. (Boeing)

vic — Display program for a Tektronix 4010 compatible device with penplot options for HP 4- and 8-pen plotters. Takes db or Caesar files. (Bruce Yanagida, Boeing; Larry McMurchie, UW/NW VLSI Consortium).

Rule Checkers

Geometric and electrical rule checking.

drc Design rule checker from Carnegie-Mellon. Checks MCSIS NMOS (buried contact) rules on Manhattan CIF designs. (Dorothea Haken, CMU)

drcscript Merges the design rule violation files created by *drc* with the CIF design file for display purposes. (Dorothea Haken, CMU)

lyra • Performs hierarchical design rule check on a Caesar-formatted design using a corner based algorithm. NMOS, CMOS and user-defined rulesets are available. (Michael Arnold, UCB)

erc Checks circuit description of an nmos design for consistent electrical properties. (Boeing)

Circuit Extractors

Extract simulation database from layout database.

mextra • Extracts a .sim file from a CIF input file. *Mextra* provides input files for *erc*, *esim*, *crystal*, *spice* and *rnl*. (Dan Fitzpatrick, UCB)

Simulation Tools

Logic and timing simulation.

spice2g6 The well known device level circuit simulator with minor mods to output an additional file that allows multiple time series plots to be made. (Lawrence Nagel, UCB)

rnl An event driven "timing" simulator. It uses logic levels and a simplified circuit model to estimate timing delays through digital circuits. It also has a mode that allows it to be used as a switch (gate) level simulator. (Chris Terman, MIT)

esim • An NMOS gate level simulator. It uses logic levels and models transistors as perfect switches. (Chris Terman, MIT)

crystal • A static timing verifier. It uses a simplified circuit model to estimate the worst case delay through a circuit. (John Ousterhout, UCB)

powest • Reads an nmos circuit description in the format required by *esim* and writes estimates of its DC power requirements.

mup Displays, a multiple time series plots on a Printronix printer, the simulation output from either *spice* or *rnl*. (William Beckett, UW/NW VLSI Consortium).

Filters and Utilities

Filters to convert from one database to another and useful utilities.

db2cif Converts from db format (produced by a PLAP program) to CIF format. (Boeing)

cif2ca • Converts from CIF format to Caesar format. (Peter Kessler, UCB)

ca2db Converts from Caesar format to db format. (Bruce Yanagida, Boeing)

netlist Generates circuit descriptions (net lists). The output is a .sim file. (Chris Terman, MIT).

presim Converts a .sim file into the binary format required by *rnl*. In the process *presim* simplifies the circuit by identifying gates in the circuit. (Chris Terman, MIT)

pspice Runs *sim2spice* and *spcpp*. In addition to running these programs it concatenates various files so as to create a complete Spice input deck. This reduces considerably the effort of simulating variations on a particular circuit. (Rob Fowler, UW/NW VLSI Consortium)

sim2spice • Reads a .sim file containing a description of a circuit and writes a .names file and a .spice file. The former contains a translation file from node names in the .sim file to the Spice node numbers. The .spice file contains a description of the

devices in the circuit in a form acceptible to Spice. (Dan Fitzpatrick, UCB)

spcpp Facilitates the writing of Spice input by allowing the user to refer circuit nodes using mnemonic node labels rather than Spice node numbers. (Rob Fowler, UW/NW VLSI Consortium)

spice A *csh* script for running *spice2g6*. (Lawrence Nagel, UCB)

APPENDIX: II

Conference on Technology Based Consortium Formation

UW/NW VLSI Consortium
Department of Computer Science - University of Washington
August 6-7, 1984 University Tower Hotel, Seattle, Washington

Program Moderator: Paul Young, Professor & Chairman
Computer Science Department, University of Washington

Monday, August 6

8:00am	Registration, check-in
9:00am	Gordon Bell, Keynote Speaker - Former VP, DEC, VP, Encore Computers, Boston, MA
9:40am	Admiral Inman - President, MCC, Austin, TX
10:20am	Coffee Break
10:50am	Frank Vince - VP, CDC, Minneapolis, MN
11:30am	Paul Penfield - Professor of EE, MIT, Cambridge, MA
12:10pm	Lunch
1:00pm	Donald Pederson - Professor and Chairman of EECS, UC Berkeley, CA
1:40pm	Larry Sumney - President, SRC, Triangle Park, NC
2:30pm	Coffee Break
3:00pm	Donald Beilman - President, MCNC, Triangle Park, NC
3:40pm	Legal Panel Auzville Jackson - President, Tennessee Technology Foundation, Nashville, TN Rob Gullette - Senior Counsel, Boeing Computer Services, Seattle, WA Carol Niccolls - Assistant Attorney General, Att. Gen. Division, UW, Seattle, WA Allen Wagner - Assistant Counsel to the Regents, Berkeley, CA
4:40pm	Movie Presentation/Computer Museum Gwen Bell - Director, Computer Museum, Marlboro, MA
5:40pm	Load busses for transportation to Virginia V (Pier 55)
6:00pm	Busses leave for Pier 55
7:00pm	Virginia V cruise/banquet
10:00pm	Return to dock

Tuesday, August 7

8:00am	Kit Bradley - Software-Hardware Engineering Manager, Tektronix, Beaverton, OR
8:40am	Ted Kehl - Professor of Computer Science, UW, Seattle, WA
9:20am	Coffee Break
9:50am	Paul Losleben - Sr. Program Manager, VLSI Program, DARPA, DoD, Washington, D.C.
10:30am	Panel on UW/NW VLSI Consortium Kit Bradley - Chairman, Board of Directors, UW/NW VLSI Consortium, (Tektronix, Beaverton, OR) Ted Kehl - Prof. of CSci. (UW, Seattle, WA) Bill Henry - Manager, VLSI CAD (Boeing, Seattle, WA) Paul Losleben - Senior Program Manager, VLSI Program, DARPA
11:20am	Martha Russell - Minnesota Microelectronics & Information Sciences Center (MEIS), Minneapolis, MN
12:00pm	Lunch
1:10pm	Auzville Jackson - President, Tennessee Technology Foundation, Nashville, TN
1:50pm	Henry Stadler - Division Technologist, Automation & Energy Systems Division Jet Propulsion Laboratory, Pasadena, CA
2:30pm	Bill Johnson - VP, DEC, Maynard, MA
3:10pm	Norm Peterson - Director, Govnmt. Com. on Science & Technology, State of Illinois
3:50pm	Close
7:30pm	China Exhibit, Pacific Science Center, 200 Second Avenue North (Tickets are $7.50 each, and may be purchased at the conference registration table.)

APPENDIX III: OUTLINE OF INTENSIVE CMOS VLSI DESIGN COURSE.

Day 1:

- Introduction to MOS technology
  - History and current state-of-the-art
  - Power dissipation
  - Implementation strategies: gate-array, standard cell, full custom
  - Fabrication yields
- Basics of design in CMOS
  - Elementary logic devices: inverters, gates.
  - Device sizing
  - CMOS latch-up phenomena
  - Simulation models of simple devices
- Guided lab exercise
  - Introduction to the Unix operating system, Emacs editor and electronic mail system
  - Spice simulation of an inverter
  - Switch level simulation of an inverter; creation of netlists and control files.

Day 2:

- Complex logic functions
  - Latches, MS-flipflops, Registers, Counters, RAM's, ROM's.
  - Dynamic versus static logic.
- Discussion of a design example:
  - 32-bit floating point unit
- Guided lab exercise
  - Construct a netlist of a 10-bit shift register
  - Construct a control file for this circuit and simulate using RNL.

Day 3:

- Circuit design with finite state machines and PLA's.
- Design for testability
  - LSSD, Bilbo, selftest
- Testvector generation and fault simulation

Day 4:

- The Isocmos fabrication process
  - Identification of mask layers
  - Design rules
  - Device sizing
- Lay-out of simple logic functions
- Introduction to layout programs CAESAR and PLAP.
- Guided lab exercise:
  - Lay-out of a latch cell using both CAESAR and PLAP.

Day 5:

- Chip lay-out planning
  - Average density observations
  - Interconnect strategies and cell placement considerations
  - Cell design
- Discussion of the lay-out of an example
- Guided lab exercise
  - Layout of a 10-bit shift register
  - Geometric design rule checking
  - Circuit extraction
  - Simulation of the extracted circuit with RNL using the control file developed during the day 2 lab exercise.

- Design and lay-out of a sequential state machine

Day 6:

- Standard cell design methodology
  - Standard cells available
  - Automated routing procedures
  - Interfacing to the standard padframe
- Guided lab exercise
  - Use the standard cells to construct another 10-bit shift register.
- Requirements of the final design:
  - A complete and succesfully simulated electrical design
  - No layout design rule errors
  - A complete and succesful simulation of the extracted circuit using the same control file used during simulation of the netlist representation of the circuit.